



DNSSEC, Explained

Signing your DNS so that a forged answer
can be told apart from a real one

A plain-English guide for people who run businesses, not DNS servers. Read it after the DNS guide. It is the one security feature in this series where getting it wrong is worse than not doing it at all, so the procedures matter more than the theory.

The problem it solves

Ordinary DNS answers carry no proof of where they came from.

When anyone types your domain name, or sends you an email, a resolver somewhere asks the DNS where your domain lives and believes the first answer that looks right. Nothing in that answer proves it came from you. An attacker who can get a forged answer in first — by poisoning a resolver's cache, or by sitting on the network in between — can send your visitors and your mail somewhere else.

The same forged answer, with and without DNSSEC.

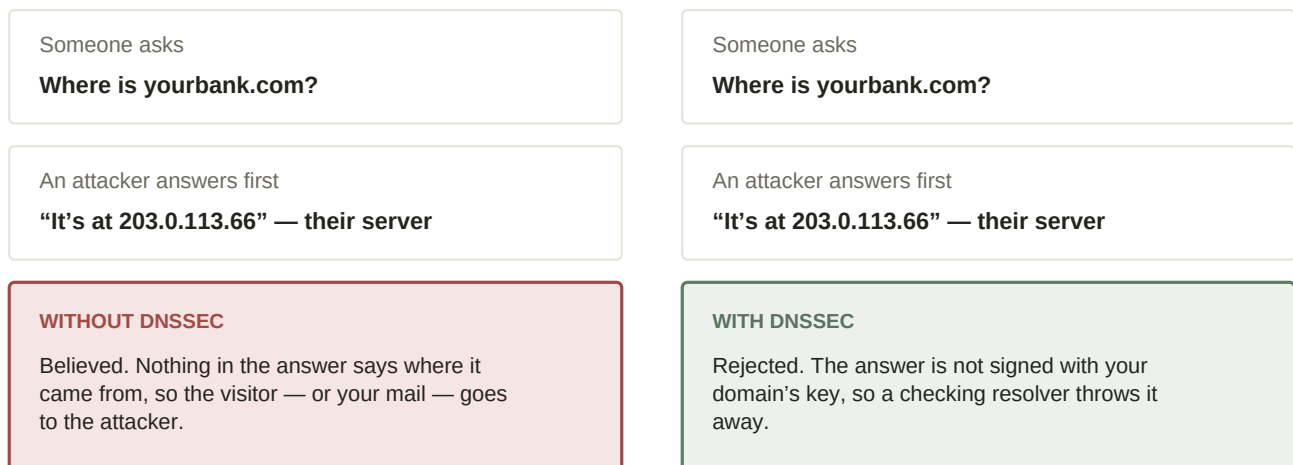


Figure 1 — DNSSEC does not stop the forgery being sent. It stops it being believed.

DNSSEC adds a signature to every answer. A resolver that checks signatures — most large public resolvers and many ISPs do — can tell a genuine answer from a forged one and discard the forgery.

What it does not do

DNSSEC does not encrypt anything and does not hide which sites anyone visits; that is a different technology. It proves authenticity, nothing more. It is also not an email standard — but DANE, the strictest way of protecting mail in transit, is built entirely on top of it.

How trust gets to your domain

A chain from the top of the DNS down to you.

Each level vouches for the one below it.

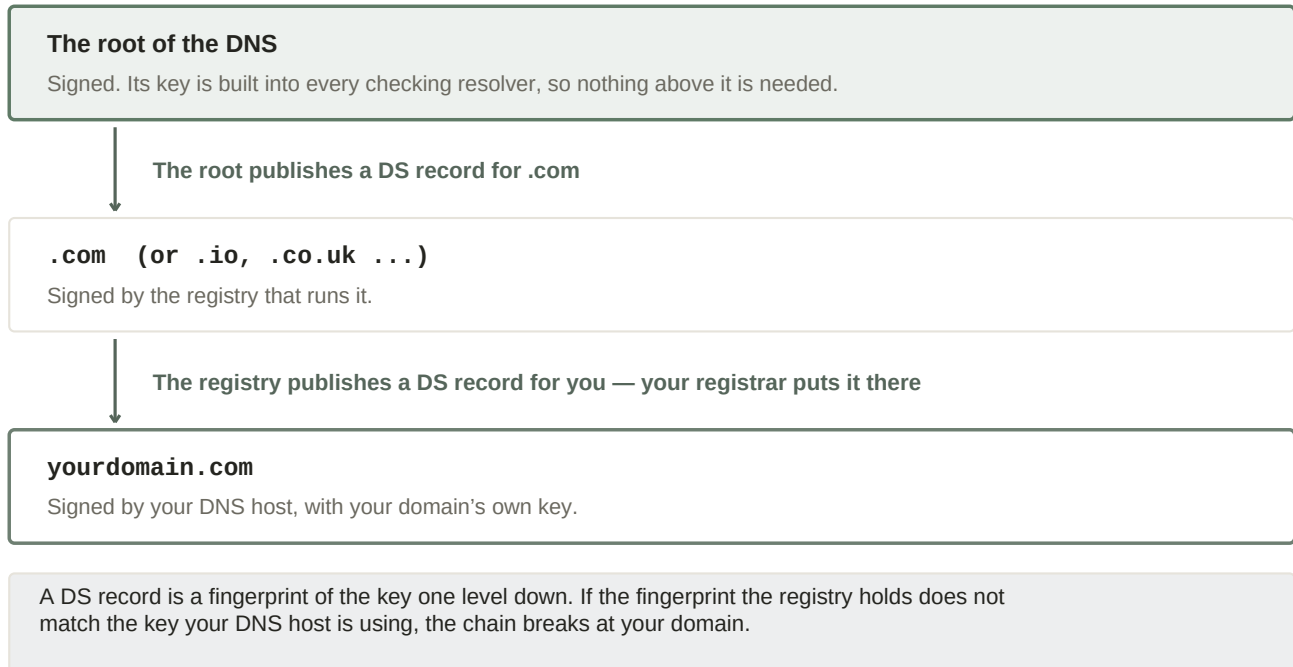


Figure 2 — The chain of trust. Your link in it is the DS record.

A checking resolver starts at the root, whose key it already trusts. The root vouches for .com by publishing a fingerprint of .com's key, called a DS record. The .com registry does the same for your domain. Your DNS host signs your records with the key that fingerprint describes. If every link matches, the answer is proven genuine; if any link does not, it is rejected.

Most top-level domains are signed, so the first two links are already in place. Whether your domain is signed depends entirely on the last link, and that is the part this guide is about.

The two halves you have to keep in step

This is the page to remember.

Two companies, two control panels, one record that has to match.

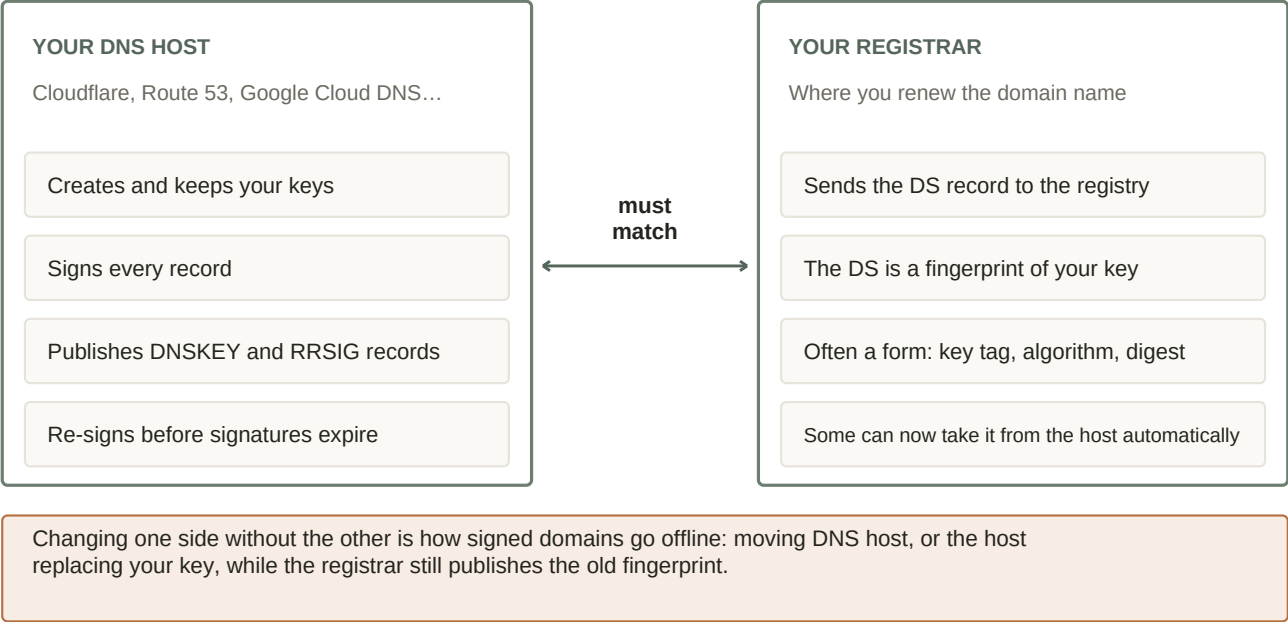


Figure 3 — Your keys live with your DNS host. Their fingerprint lives with your registrar.

For many businesses these are the same company, and then most of this happens behind a single switch. When they are different companies, DNSSEC is configured in two places, and the two must describe the same key. Nearly every DNSSEC outage is the two halves disagreeing — usually after a change of DNS host.

The records you will see

Four record types, none of which you write by hand.

Record	Where it lives	What it is
DNSKEY	Your zone, at your DNS host	Your domain's public keys. Usually two kinds: a key-signing key (flag 257) that the DS points at, and a zone-signing key (flag 256) that signs everything else.
RRSIG	Your zone, beside every record	A signature. Each has a start and an end date, and your host re-signs before the end — every few days at most hosts.
DS	The registry, via your registrar	The fingerprint of your key-signing key. The only part that is not in your zone, and the part that switches DNSSEC on.
NSEC / NSEC3	Your zone	Signed proof that a name does <i>not</i> exist, so an attacker cannot forge "no such domain" either.

What a DS record looks like

```
yourdomain.com.  IN DS  12345 13 2 3A9F0C61D2E847B5...
                    |      |  |  |
                    |      |  | digest: fingerprint of the key
                    |      |  | digest type: 2 = SHA-256
                    |      |  | algorithm: 13 = ECDSA P-256
                    |      |  | key tag: which key it describes
```

Your DNS host shows you these four values. Registrars ask for them either as one line like this or as four separate boxes. The key tag and algorithm must match a key your host is actually publishing, and the digest must be that key's fingerprint — a checker can confirm both.

Why broken is worse than unsigned

The most important page in this guide.

Broken is worse than not signed at all.



Figure 4 — A security feature that fails closed.

An unsigned domain is in the same position as most of the internet: it works, it just cannot prove its answers. A domain whose DS record no longer matches its keys, or whose signatures have expired, is something else. Checking resolvers are required to reject what fails, and they cannot tell your mistake from an attack, so they refuse to answer at all. Your website and your inbound mail disappear for everyone using them, while the people who test it from a non-checking network see nothing wrong.

What breaks it, in practice

Changing DNS host while the registrar still publishes the old DS. A host that stops signing, so signatures expire days later. A key replaced at the host without the DS being updated. All three are the two halves disagreeing, and all three are avoided by the procedures on the next pages.

Which algorithm

Your DNS host chooses; you should know what it chose.

The algorithm numbers your DNS host may show you, and what to make of them.

15	Ed25519	Recommended. Small, fast, modern.	RECOMMENDED
13	ECDSA P-256	Recommended. The most widely used today.	RECOMMENDED
8	RSA / SHA-256	Still recommended and universal, but its much larger answers are a reason to move to 13 or 15.	RECOMMENDED
14, 16	ECDSA P-384, Ed448	Allowed. Fine, less common.	MAY
10	RSA / SHA-512	Not recommended for new signing.	NOT RECOMMENDED
5, 7	RSA / SHA-1	No longer to be used for signing since 2025. Still accepted for now — plan a change.	MUST NOT
1, 3, 6, 12	MD5, DSA, GOST	Obsolete. Checking resolvers ignore them, so the domain is effectively unsigned.	MUST NOT

The DS record has its own choice of fingerprint: SHA-256 (type 2) is the one to use. SHA-1 (type 1) is now MUST NOT.

Figure 5 — Current guidance, from the IANA DNSSEC algorithm registry.

If your host offers a choice, pick 13 or 15. Both are recommended, supported by every major checking resolver, and produce answers small enough to travel without problems. Algorithm 8 is still fine; it is simply larger. If you find 5 or 7, the SHA-1 algorithms, ask your host to change them: since late 2025 they are no longer to be used for signing. They still validate today, so this is a planned change, not an emergency. Changing algorithm is a key change your host carries out; it should not involve you beyond updating the DS.

One more setting you may see reported is NSEC3 "iterations" and "salt". Current guidance is zero iterations and no salt — written 1 0 0 -. Anything else costs resolvers work for no real benefit. It is your host's setting, and worth a mention to them, not a reason to worry.

Turning it on

Four steps, in an order that cannot break anything.

Switching DNSSEC on, in the order that cannot break anything.

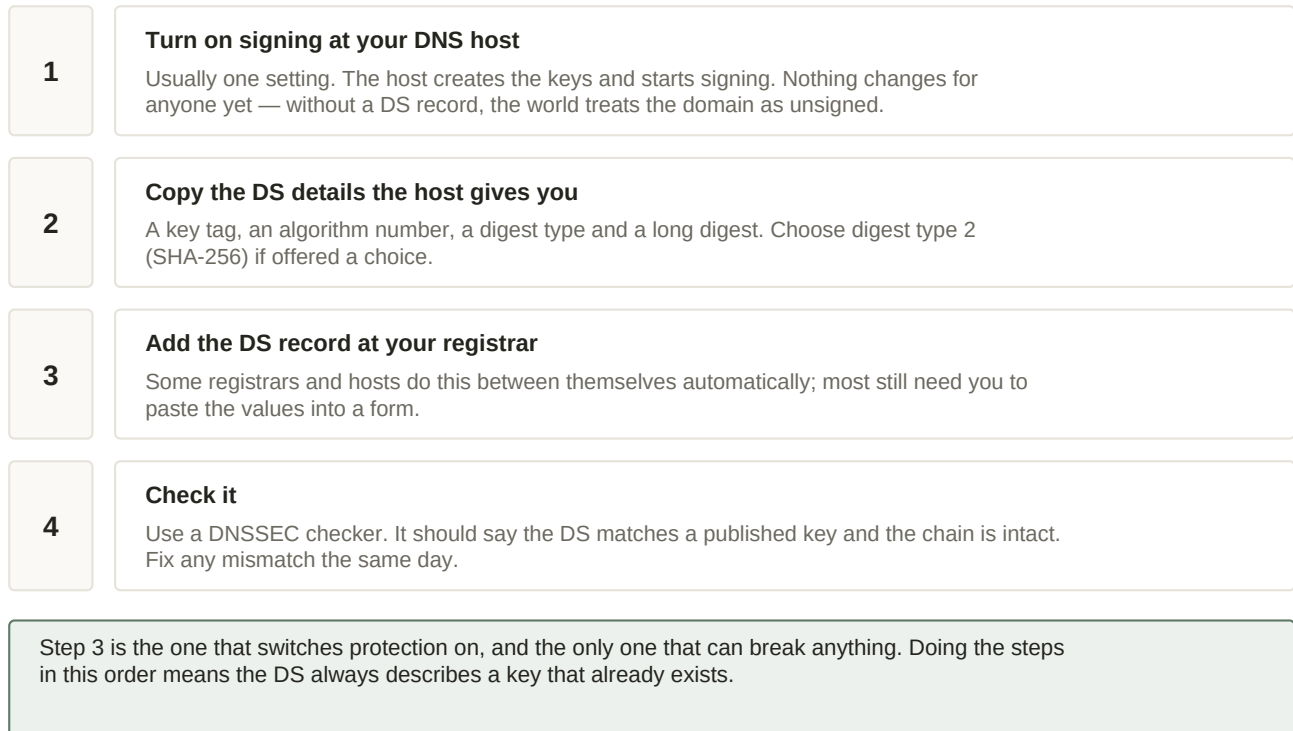


Figure 6 — Keys first, then the fingerprint that points at them.

If your DNS host and registrar are the same company, steps 2 and 3 usually happen on their own when you flip the switch. Check afterwards anyway: "enabled" in a control panel and "secure" from the outside are not always the same thing.

Changing DNS host

The moment most signed domains go dark.

Moving to a different DNS host without disappearing.

1

Remove the DS record at your registrar

This turns DNSSEC off as far as the rest of the internet is concerned. Your domain keeps resolving, just unsigned.

2

Wait for the old DS to be forgotten

Resolvers remember a DS record for its time-to-live, set by the registry — commonly a day, sometimes two. Wait at least that long.

3

Move the domain to the new host

Change nameservers at the registrar as normal. Nothing about DNSSEC can break now, because nothing is expecting it.

4

Turn signing on at the new host and publish its DS

The same as switching on for the first time: new keys, new DS, then check.

Skipping step 2 is how most DNSSEC outages happen: the registry still vouches for the old host's key while the new host signs with a different one. Some providers can transfer signing without a gap — ask both before you start, not halfway through.

Figure 7 — Switch DNSSEC off, wait, move, switch it back on.

This gives up DNSSEC protection for a day or two, which is a small price against the alternative. The faster route — keeping the domain signed throughout by having the old and new hosts publish each other's keys — exists, but needs both providers to support it. Ask before you start.

A short checklist

Before, during, and every time something changes.

Before you turn it on

- **Know who your DNS host and your registrar are.** If they are different companies, you will be using both control panels.
- **Check what your DNS host offers.** Most major hosts sign with one click; a few do not support DNSSEC at all.
- **Prefer algorithm 13 or 15** if you are given a choice, and SHA-256 (type 2) for the DS.

Keeping it working

- **Write down that DNSSEC is on**, and where the DS is. The risky change is the one made two years from now by someone who was not there.
- **Never change nameservers without removing the DS first** and waiting — Section 08.
- **Check after any change** at either the host or the registrar.

Signs something is wrong

- Some people can reach your website or send you mail and others cannot, depending on which network they are on.
- A checker reports that no key matches the DS, or that signatures have expired. Treat either as urgent: the domain is failing for checking resolvers right now.
- Your DNS host shows DNSSEC enabled, but a checker reports the domain as unsigned. The DS has not been added at the registrar.

Check your own domain

ReaperCloud's DNSSEC tool follows the chain from the registry to your keys, confirms the DS fingerprint actually matches, checks signature dates and the algorithms in use against current guidance, and asks a checking resolver whether it accepts the domain. `reapercloud.com/dnssec.php`.

This document describes DNSSEC as specified in RFC 4033, RFC 4034 and RFC 4035, with NSEC3 per RFC 5155 and RFC 9276, and algorithm guidance from the IANA DNSSEC registries as defined by RFC 9904 and updated by RFC 9905. It is an explanation, not a specification: where they differ, the RFCs are correct and this is simplified.